Telepresence for In-situ Microscopy

B. Parvin, J. Taylor, B. Crowley, L. Wu, W. Johnston
Imaging and Distributed Computing Group
Information and Computing Sciences Division
Lawrence Berkeley National Laboratory
Berkeley, CA 94720

D. Owen, M.A. O'Keefe, U. Dahmen
National Center for Electron Microscopy
Material Sciences Division
Lawrence Berkeley National Laboratory
Berkeley, CA 94720

ABSTRACT

We present an approach for remote operation of instruments in the Internet environment. We have applied this approach to in-situ electron microscopy experiments that require dynamic interaction with the specimen under observation, as it is excited with external stimuli, i.e., temperature variation, EM field variation, etc. The dynamic operations include control of the sample's position and orientation under the electron beam, the illumination conditions and focus, etc. Remote control via wide area networks like the Internet that do not offer real time data and command delivery guarantees is not practical for the finely tuned adjustments that dynamic studies require. We suggest that an effective approach to remote operation for this class of dynamic control applications must involve *automated* control operations performed *near* the instrument in order to eliminate the wide area network real-time delivery requirement. Our approach to this problem is based on advanced computer vision algorithms that permit instrumentation adjustments to be made in response to information extracted from the video signal generated by the microscope imaging system. We have determined the type of servo loops needed to enable remote operation and collaboration, and have introduced a novel partitioning of the control architecture for implementing this approach. In this partitioning, the low frequency servo loop functions that require direct human interaction are performed over the wide area network, and those functions that require low latency control are performed locally using the automated techniques. This approach hides the latencies in the wide area network and permits effective remote operation. The result is telepresence that provides the illusion of close geographical proximity for in-situ studies. Our testbed is a 1.5 MeV transmission electron microscope, which can now be used on-line via the global Internet.

1 Introduction

The current trend in applying multimedia capabilities to laboratory environments aims at the routine use of video image sequences over a wide area network. The issues here are the ability to collect, store, compress, and display video sequences for users who each might have critical bandwidth requirements for a finite resource [10]. The next natural evolution of multimedia systems is to provide the computational framework to analyze images, extract meaningful information from a video sequence in real-time, and provide the ability to *manipulate* experiments based on the content of the video sequence, over the wide area network. One such application of the approach is a system for remote operation of in-situ microscopy. Our testbed is a 1.5 million electron volt (MeV) transmission electron microscope (HVEM), shown in figure 1, that is operated by the National Center for Electron Microscopy.

In-situ microscopy refers to a class of scientific experiments in which a specimen is excited by external stimuli. The stimuli could be in the form of temperature variation or stress in the sample environment The interaction of the external stimuli and specimen can result in sample drift, shape deformation, changes in object localization, changes in focus, or simply anomalous specimen responses to normal operating conditions. Currently, during the in-situ experiments the operator must make constant adjustments to the instrument to maintain depth of focus and compensate for various drifts. These are labor intensive and error prone tasks –requiring a high bandwidth video link to the operator– that are nearly impossible to do in wide area networks due to limited network bandwidth. Although researchers have built telepresence systems for electron microscopy, these systems do not address the more complex issues of in-situ microscopy. The novelty of our system lies in the approach that uses automation techniques for on-line quantitative analysis, and manipulation and compensation of the necessary experiment parameters. Thus, by relieving the operator of having to do the dynamic adjustment of the experimental setup, remote collaboration and remote operation of the in-situ studies over a wide area network are made possible. This is accomplished through the use of advanced computer vision techniques to provide automation for microscopy applications [14, 15]. From this perspective, we have defined and developed a series of computational techniques that are necessary for remote in-situ applications. These include 1) image compression, 2) autofocusing, 3) self calibration, 4) object detection, 5) tracking using either high level or low level features, and 6) servo-loop control mechanisms. These functions are implemented over a distributed client-server software architecture for better throughput, scalability and modularity.

In the next sections, we outline the approach and computational platform for remote operation, together with some of the details of the instrumentation of the HVEM. In section 4, we summarize the algorithms that are used to realize the functions for remote in-situ experiments. Then in section 5, descriptions of the software architecture and the performance parameters are outlined. Finally, we conclude with the limitations of the system and directions of

[*] This work is supported by the Director, Office of Energy Research, Office of Computation and Technology Research, Mathematical, Information, and Computational Sciences Division, and Office of Basic Energy Sciences, Material Sciences Division of the U. S. Department of Energy under contract No. DE-AC03-76SF00098 with the University of California. The LBNL publication number is 38548. E-mail: parvin@george.lbl.gov

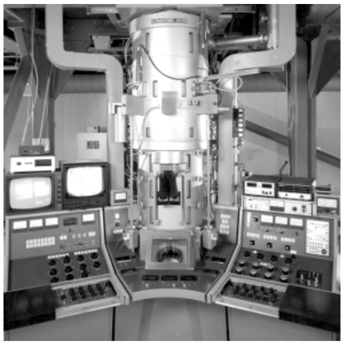


Figure 1: The high voltage electron microscope at NCEM

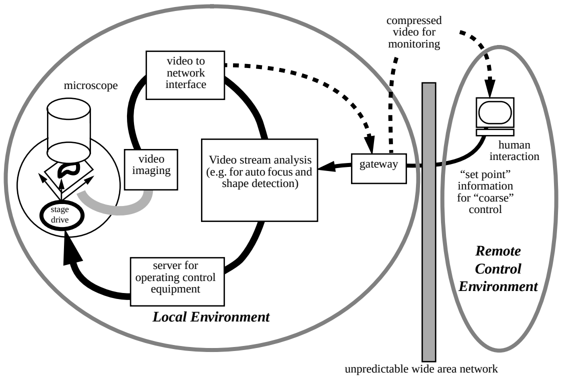


Figure 2: Remote operation architecture

the future efforts.

2 Approach

Our general approach to the problem of remote control of dynamic experiments is to separate the basic human interaction of establishing control system parameters like gross positioning, identifying objects of interest, etc. (which do not require low latency communication) from the control servoing that performs operations like auto-focus, object detection, continuous fine positioning due to thermal drift, etc., which do require low latency communication.

The human interaction operations, together with the supporting human communication involving video and audio teleconferencing, can easily be performed in a wide area network environment [9, 12].

The dynamic control operations must occur in a much more controlled environment where the control operation and the monitored response to the control or stimuli have to be coupled by low latency communication that is not possible in wide area networks. For these operations, we use computer vision techniques to provide the monitoring of the response to the control operations by extracting position and shape information from the video imaging output of the microscope.

This image content analysis, and the dynamic control resulting from the information that is fed back to the control system, is automated and performed in a local environment. That is, the computers that acquire and analyze the video images and then communicate with the control system are all connected by fast local area networks. The approach is illustrated in figure 2.

3 Computational platform

The computational platform that implements the automated control in the local environment must be able to acquire images, process them at the required bandwidth, and manipulate a large number of functions for operating the HVEM. Our strategy for partitioning the required operations is based partly on design philosophy, i.e., scalability, modularity, and cost, and partly on the availability of data acquisition components (DAC) for various hardware platforms. For these reasons, a Sun Microsystems workstation is used for image capture, a Digital Equipment Corporation (DEC) symmetric multiprocessor is used for CPU intensive operations, and a PC is used for data acquisition. Physically, the Sun and PC are operating near the microscope, while the DEC is located in another building of the Laboratory, but is still connected via a LAN. The Sun and DEC are on a FDDI ring (100 Mb/s) for high speed image transfer as shown in figure 3. In this configuration, the local Sun workstation is mainly used for testing and on-line quantitative analysis by local users.

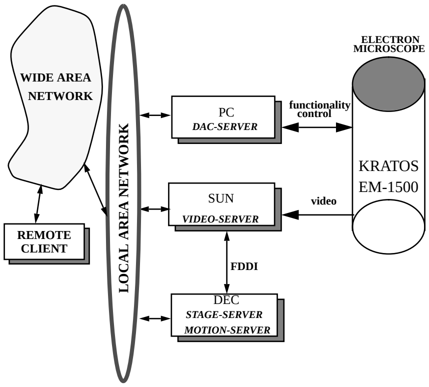


Figure 3: Computational infrastructure for remote in-situ microscopy. The italic names specify the servers that run on each platform.

3.1 Microscope Instrumentation

The HV electron microscope possesses a large number of knobs and switches, not all of which are necessary or appropriate for teleoperation. We have placed those functions that allow *safe* remote operation of the instrument under computer control. For example, control of the filament current–used for generating electrons–is a feature that is not offered to remote users because a novice operator (or intruder) could accidentally damage the filament despite the microscope's safety features. Critical functions can be adjusted only by the local operator. The remotely operated functions have limit switches to prevent remote users from going beyond safe boundaries.

Teleoperation over the wide area network is achieved with 3 DAC boards and one stepper motor controller board that are integrated into the PC. Current remote functions include translating a specimen in the X and Y directions, tilting the specimen, altering focus, and controlling the temperature. Other functionalities[1] to be added in the near future include starting a session, ending a session,

[1] Will be Supported by the existing DAC boards.

altering the magnification, changing the beam size, shifting the beam, controlling the position of the aperture, and generating a high resolution micrograph.

4 Computational techniques

In the context of remote in-situ microscopy, the system must provide the look and feel available to the local operator, and hide the inherent latency in the wide area network. The look and feel is achieved through an appropriate user interface. The hiding of the network latency is achieved through *visual servoing*. Visual servoing is the process of interpreting the video data and manipulating various control mechanisms based on the image content. We have developed a collection of computer codes for image interpretation that can be used for closing the servo-loop in the local area network. These include image compression, autofocusing, thermal drift correction in the absence of known geometric shape features, self calibration, object detection and tracking, and close loop control.

4.1 Image compression and autofocusing

Both image compression and autofocusing use the wavelet transform as their underlying principle. We use Daubechies kernels [8] that are simple, orthogonal, highly localized, and separable for two dimensional processing. The main advantage of the wavelet transform is that it can represent local feature activities at multiple scales through spatial decimation. During image compression, the low order wavelet coefficients are ignored and the remaining ones are encoded in blocks of 16-by-16 pixels. The remote user has full control over what percentage of wavelet coefficients are used for compression. We also, provide delta encoding as another option for image compression. The approach is similar to DPCM (differential pulse code modulation); however, certain artifacts due to binary delta function of the DPCM are inhibited.

Autofocusing has two modes of operation: initialization and run-time.[2]

The goodness of the focus is measured by the sum of the wavelet coefficients. During the initialization –with the aperture in the "out" position– we search for a lens current that minimizes the sum of the wavelet coefficients. This corresponds to a search for a flat field, i.e., minimum contrast. And during the run-time –with the aperture in the "in" position– we search for a lens current that maximizes the sum of the wavelet coefficients. This corresponds to a search for focal position when highly diffracted components of the beam are blocked by the objective aperture. This is the same focusing protocol that microscopists use for transmission electron microscopy. The difference between the initialization and run-time mode is based on the range of currents that are tested for optimum focus position. During the initialization mode, the best focus position is obtained through a binary search over a large range of current values. At run-time –as the specimen is heated– small adjustments are made in the focal position to compensate for 3D changes of the precipitate position.[3]

4.2 Drift control and self calibration

Thermal drift control and self calibration use the optical flow field estimation as their undelaying principle. Drift control involves motion estimation, which is a necessary component of fixing on a moving object. Once the motion between consecutive frames is known, the stage assembly where the object resides, is re-positioned to compensate for the motion. In retrospect, the motion corresponds to a continuous *flow* of the image world across the retina and serves as an underlying perceptual cue for higher level cognitive processes. From a computational perspective, the flow is essentially the instantaneous velocity of each pixel in the image [2, 3, 4]. The general solution to the velocity field is inherently under-constrained because the number of equations that define the flow field variables is insufficient. However, by exploiting the nature of the flow field in the image, or by using an appropriate bank of filters, a constrained or over-constrained system of equations can be constructed. In the case of thermal drift during in-situ electron microscopy, the nature of the *global* motion provides the mechanism for constraining the flow equations. This constraint is expressed by the fact that the motion between consecutive frames is *affine*. Hence, a least-squares solution with respect to the parameters of the affine transform can be constructed. Our current implementation runs at 4 Hz on the DEC multiprocessor, which is sufficient for maintaining thermal stability for an in-situ electron microscope. The details of this technique are provided in Appendix A, and an example of motion estimation is shown in figure 4.

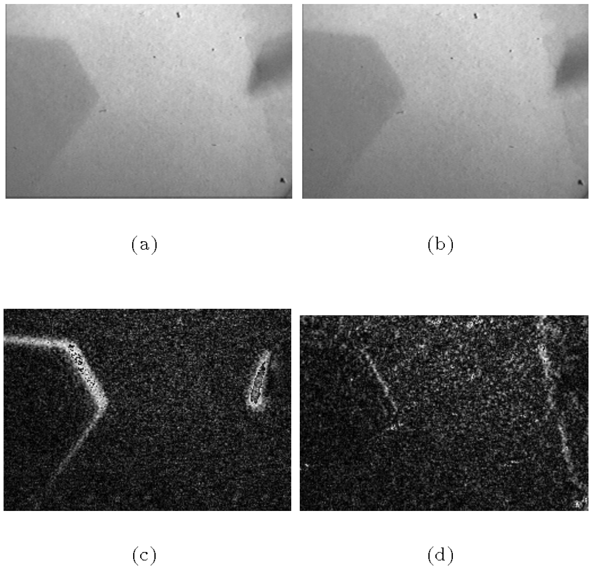


Figure 4: Motion estimation by optical flow field: (a) frame 1; (b) frame 2; (c) raw difference; (d) raw difference after applying the computed affine transform.

The estimation of the flow field is also used for self-calibration. The sample position is controlled with an XY stepper motor controller. Hence, it is necessary to map the pixel size as a function of magnification into the number of steps in the sample positioning system. This is accomplished by having the stepper motor make a controlled motion–as a function of magnification–and computing the flow field from two consecutive frames.

4.3 Detection and Tracking

The main objective for the in-situ microscopy experiments that are being addressed by this work is to observe the

[2] The focus is adjusted by a 16-bit D/A converter that controls the lens current.

[3] Precipitates are the principal objects of the in-situ experiments, and the foil carrying the precipitate buckles under heat stress.

change in the shape of precipitates[4] as the temperature is increased and decreased. We have developed techniques for detecting these precipitates, tracking them, and then use the result of tracking to correct for thermal drift. It turns out that precipitates have convex geometrical representations that may also satisfy other constraints such as parallel or circular symmetries. The observed images are generally noisy, have poor contrast, and, depending on the position of the electron beam and the foil angle, suffer from shading artifacts. We have recently developed a technique for detecting convex objects from the scene based on the perceptual grouping principles [5, 11, 13, 15, 17]. The approach relies on grouping line segments –obtained from Canny's edge detector [7]– to form convex sets. This is achieved through an efficient global convexity test on groups of line segments in conjunction with a dynamic programming search strategy [15]. An example of convex precipitate and convex shape detection is shown in Figure 5.

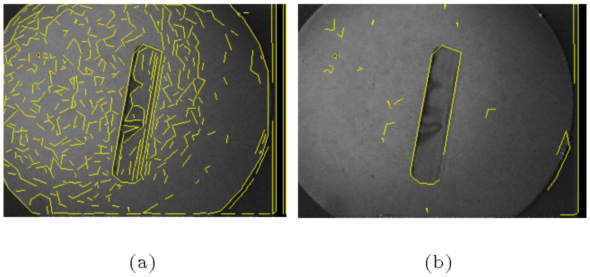


Figure 5: Automated detection of a convex precipitate: (a) original image with edges overlaid; (b) detected convex sets.

The detection system provides a coarse description of the precipitate in the form of bounding polygons. This description is then refined and tracked in subsequent frames using an architecture that was discussed in our recent paper [14]. In this context, detection of precipitates occurs only in the first frame for the purpose of initialization. The contour refinement algorithm [5] uses a cost function that is optimized through dynamic programming. In this context, both detection and tracking use dynamic programming at different scale of hierarchy. The cost function encodes the desirable properties of the refined contour in terms of high and low level feature activities. The low level features refer to pixel level information, such as local edge magnitude and direction. In our system the high level constraints, derived from the bounding polygon, affect the contour refinement in two ways. These include geometric constraint and the scope of the search. This is accomplished by smoothing the initial polygon with a Gaussian kernel and bounding the refined contour to lie in a small neighborhood as defined by the normal lines to the smooth curve. The rationale for Gaussian smoothing is that the bounded polygon is not smooth and the normal lines may not intersect the actual boundary of the precipitate. However, by smoothing the bounding polygon, the normal lines scan the precipitate along its real boundary smoothly. The details of the technique are summarized in appendix B.

An example of tracking of precipitate during the heating and cooling cycle is shown in Figure 6. Note that the precipitate becomes round and then facets as the temperature is increased and then decreased. We use a multigrid implementation of the above algorithm for maximum speed up and higher tolerance for large motion. The algorithm has performed well in the presense of shading, noise, nonuniform illumination, and reduced contrast. The system automatically tracks the shape, controls the drift, and hides the network latencies from the remote user. The drift control is based on tracking and compensating for the centroid of the contour. This is shown in figure 7, where the centroid is shown with a cross-hair on the reconstructed image, and the direction of the motion is shown with an arrow. In addition to the topological changes in the shape, during the heating and cooling experiment, thermal drift reverses its direction as well, which is also reflected in the figure.

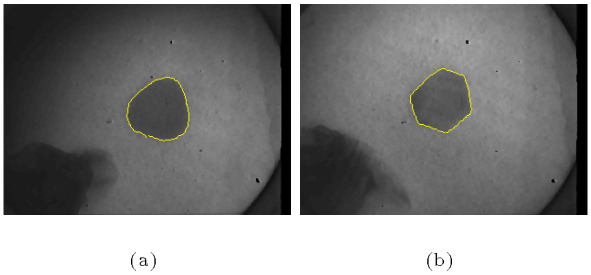


Figure 6: Tracking of a deformable precipitate during in-situ microscopy quantifies its shrinkage rate and how it facets: (a) Tracking of the precipitate during heating phase; (b) Tracking of the precipitate during the cooling phase.

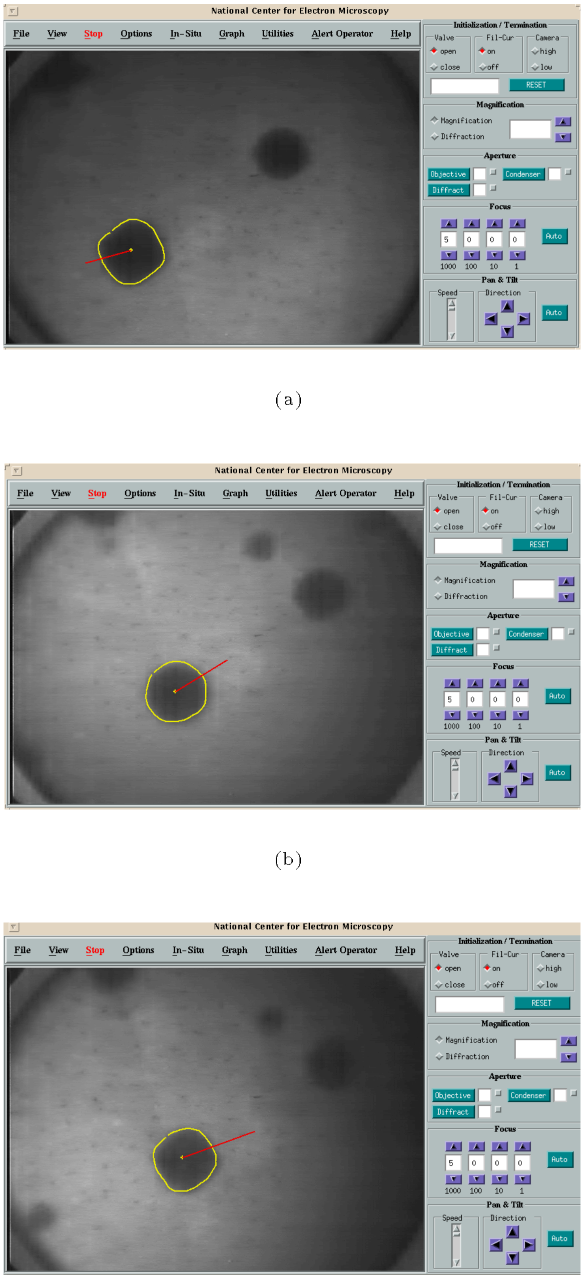
4.4 Servo control

The novelty of the servo loop mechanism is twofold. These include 1) the architecture for servicing the local and and remote requests, and 2) the use of statistical techniques for close loop servo control. Manipulation of the microscope functions may be initiated either from remote user under manual control, or from the tracking algorithms described above. In other words, coarse manipulation of the microscope is performed over the WAN, while the refined and predictable manipulation is performed over the LAN. The stage-server –running on the DEC platform– acts as a switch to arbitrate between local versus remote requests. Furthermore, by limiting the remote client interaction to one computational platform, we have limited the user authentication problem to that platform only.

The second component of our work for closing the servo loop is based on the fact that the motion of precipitate is smooth. We use a Kalman filter model to predict motion parameters from noisy measurements. Kalman filtering has been used extensively for smoothing, filtering, and prediction as reported in the literature [6, 18]. In general, the model provides smooth compensation for drift and shape tracking coupled with high tolerance for larger speed. Our implementation uses position and velocity to represent the internal state of the precipitate. In this context, the model is used to predict the trajectory of the motion. As a result, instead of making incremental correction to the XY stage platform, we place the stepper controller at a constant speed in the direction opposite to the thermal drift.

[4] Alloys of germanium or lead are used in our experiments.

[5] In our system, the functionality of refinement and tracking is achieved with the same algorithm.



(a)

(b)

(c)

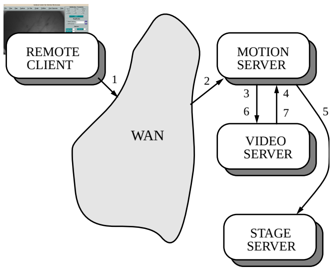
Figure 7: Tracking and compensating for drift during heating and cooling cycles. Note that direction of drift is reversed as the specimen is cooled: (a) Precipitate is initially faceted as it is heated. (b) Precipitate becomes round at high temperature and thermal drift reverses direction as the specimen is cooled. (c) Precipitate becomes faceted again at low temperature.

The speed is then refined at the tracker sampling interval. The detail of the Kalman filtering model is given in Appendix C.

5 Software architecture

The software architecture follows a distributed client-server model for scalability, performance, and modularity. There are four servers that can interact with each other in the architecture shown in figure 3. These are the i) video-server, ii) motion-server, iii) stage-server, and iv) DAC-server. The video-server –running on the Sun– captures images and transfer them in their entirely or partially to the motion-server. The motion-server –running on the DEC– manages all the image analysis and servoing. These modules are executed asynchronously and use a threads programming paradigm for parallel decomposition. The stage-server -running on the DEC– handles all the manual interaction between the remote user and the electron microscope, i.e., changing magnification, shifting the beam, etc. The DAC-server –running on the PC– reads and writes into the data acquisition components for a desired function. The DAC-server uses remote procedure calls for communication and the remaining servers use data streams through sockets for minimum delay. We now give two examples of the dynamics aspects of the motion-server.

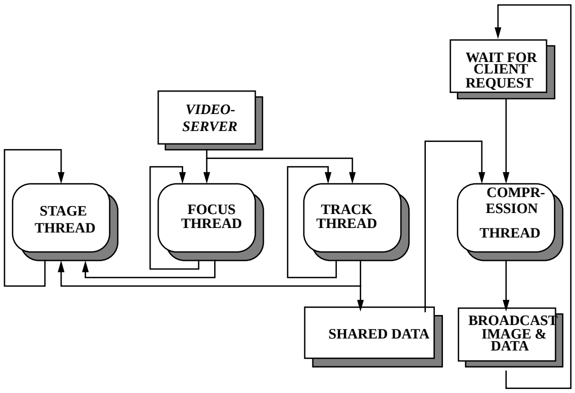
The first example is self-calibration. The details are shown in figure 8, where the interaction between different computational platforms is shown. The steps are as follows: 1) the remote client makes a request for self calibration; 2) the request is transferred to the motion-server; 3) the motion-server requests a frame from the video-server; 4) the video-server sends an image to the motion-server; 5) the motion-server requests a translation from the stage-server; and 6,7) the motion-server requests another frame from the video-server. Self-calibration involves solving a linear system of equations that provide a mapping between the pixel size and the corresponding number of stage steps. This value is retained for subsequent drift corrections.



(a)

Figure 8: State diagram for self-calibration.

A critical component of the system is in the design of the motion-server. This server has four threads that run asynchronously as shown in figure 9. The stage-thread handles all the interaction with the stage-server, and it has been isolated for modularity and higher throughput. Average time for most interaction with the PC is about 7ms. The tracking thread operates at 5-8 Hz depending to the size of the precipitate, and runs with a concurrency of two. The compression-thread runs at 1.4 Hz over the shared data, and the focus-thread runs on a single thread over the target region when the tracking thread is inactive.



(a)

Figure 9: State diagram for motion-server indicates four distinct threads that run asynchronously.

6 Conclusion

An approach and implementation for telepresence for in-situ microscopy is outlined. At the macro level, the main benefit of the proposed effort is the increased utilization of a sophisticated instrument that has restricted access due to its sensitive components and demand for operator skills. Secondary benefits include a reduction in cost associated with conducting individual experiments and an increased ability for experimenters to collaborate. At the micro level, the benefits include providing *novel generic techniques* for 1) manipulation of real-time video and 2) servo techniques for dynamic handling of video sequences –tools essential for real-time collaborative activities. The testbed for our experiment has been a 1.5 MeV Transmission Electron Microscope that is primary used for in-situ studies.

There are several limitations in the exiting architecture and algorithms. The tracking technique can be confused in the presense of artifacts such as bend contours, which are manifestation of stress –as a result of heat– on the specimen. The architecture of figure 9 is client driven, and as a result, it will not scale well to multiple users. Finally, there is no authentication built in to the current gateway, which must be an integral part of any remote instrumentation. These are current topics of our research, and new results will be shortly demonstrated in near future.

A Optical flow equations

In this section, the details of the optical flow algorithm with the affine constraint is summarized. Let the image at time $t + \Delta t$ be defined as:

$$f(x+\Delta x, y+\Delta y, t+\Delta t) = f(x,y,t) + \frac{\partial f}{\partial x}dx + \frac{\partial f}{\partial y}dy + \frac{\partial f}{\partial t}dt \tag{1}$$

With the brightness constancy assumption and some algebraic manipulation, we have

$$-\frac{\partial f}{\partial t} = f_x U + f_y V = (\nabla I)^T \cdot \begin{pmatrix} U \\ V \end{pmatrix} \tag{2}$$

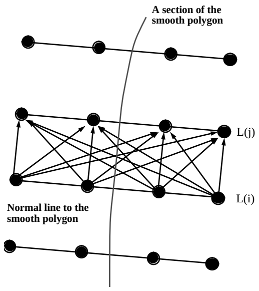

A section of the smooth polygon
L(j)
L(i)
Normal line to the smooth polygon

Figure 10: Possible paths for each point on a normal line segment

Where f_x and f_y correspond to the spatial image gradient in the x and y directions. U and V correspond to the velocity components along each axis. Constraining the flow equation into the affine motion of

$$\begin{array}{l} U(x,y) = a_1 + a_2 x + a_3 y \\ V(x,y) = a_3 + a_4 x + a_5 y \end{array} \tag{3}$$

We arrive at a well defined system of equations that is solved using the gradient descent method.

$$E(\partial a) = \sum_x \left(\Delta I + (\nabla I)^T \cdot \begin{pmatrix} U \\ V \end{pmatrix} \right)^2 \tag{4}$$

In our implementation, we use a pyramid representation of the data for the coarse-to-fine motion estimation. The main advantages include estimation of large shifts in the image plane, coupled with higher computational throughput.

B Contour refinement

In this section, the details of contour refinement and tracking is covered. Conceptually, each point on the refined contour should have high gradient and good continuity with both the low and high level features. This is reflected in the cost function in terms of making a decision for a particular path. The cost function indicates the accumulation of making a decision to link two points from two consecutive normal line segments as shown in Figure 10. Let:

1. g_i be the gradient magnitude for a point p_i on line L_i,
2. θ_i be the local edge direction at location p_i. We define the local edge direction as the direction that is normal to the direction of maximum rate of change of the local gradient,
3. $\Delta\theta_{ij}$ be the difference between the direction of gradient for points p_i and p_j,
4. γ_i be the angular difference between local edge direction and local direction of high level constraint. This is obtained by representing a local segment of the high level constraint, expressed by the smooth polygon, as a vector, and imposing that this vector should be *co-directional* with the local edge direction along the corresponding normal line.
5. η_{ij} be the angle between the local edge direction at point p_i and the vector connecting point p_i to p_j,
6. w_{ij} be the distance between points p_i and p_j located on two consecutive normal line segments, and

7. α_{ij} be the directional deviation between two consecutive points defined as $MAX\ (\Delta\theta_{ij}, \eta_{ij})$.

We formulate the cost function in such a way that the gradient is maximum, directional differences are small and the deviation for a particular path from the bounding polygon (high level constraint) is also small. In other words, the desired path should maximize the following cost function:

$$MAX_{i,j,k} \sum_{k} \sum_{i} \sum_{j} \frac{1}{w_{ij}} g_i e^{-tan(\frac{\gamma_i}{s})} \cos\left(\alpha_{ij}\right) \qquad (5)$$

where k is the length of the contour, i and j are the pixel locations on normal line to the smooth polygon, and s is a parameter that controls the amount of deviation from the high level contour (it is set to "2" in our program). This formulation indicates that: i) when $\gamma_i = \pi$,[6] the local cost function goes to zero, i.e., the local edge direction has an opposite direction to the direction of the smooth curve from the high level constraints, ii) if $\alpha_{ij} > \frac{\pi}{2}$, then the local cost will be negative and this particular sub-path will be inhibitive in the search process, and iii) by expressing the cost function as a product of internal and external forces, as opposed to the sum, we have eliminated weighting coefficients. The above cost function is optimized with the dynamic programming principle [1, 16, 14].

C Kalman model

The state space representation of the system is as follows:

$$\begin{aligned} X(k+1) &= AX(k) + n_x(k) \\ y(k) &= CX(k) + n_y(k) \qquad (6) \end{aligned}$$

Where X and y correspond to the state and observation vector, n_x and n_y are noise vectors, and A and C are constant matrices. The state vector consist of position and velocity components. In this context, the above set of equations can be rewritten as:

$$\begin{aligned} x(k+1) &= x(k) + v(k) + n_x(k) \\ v(k+1) &= v(k) + n_v(k) \\ y(k) &= x(k) + n_y(k) \qquad (7) \end{aligned}$$

Then, the filter equations will be:

$$\begin{aligned} Gain: \quad & K(k+1) = \\ & P(k+1|k)C^T[CP(k+1|k)C^T + R]^{-1} \qquad (8) \\ Where \quad & P(k+1|k) = \\ & AP(k|k)A^T + Q^* \qquad (9) \\ Update: \quad & \hat{X}(k+1|k+1) = \\ & \hat{X}(k+1|k) - K(k+1)[C\hat{X}(k+1|k) - y(k+1)] \\ & P(k+1|k+1) = \\ & (I - K(k+1)C)P(k+1|k) \qquad (10) \end{aligned}$$

References

[1] A. Amini, T. Weymouth, and R. Jain. Using dynamic programming for solving variational problems in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 855–867, 1990.

[2] J. Barron, D. Fleet, S. Beauchemin, and T. Burkitt. Performance of optical flow techniques. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 236–242, 1992.

[3] J. Bergen, K. Anandan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *European Conf. on Computer Vision*, pages 236–252, 1992.

[4] J. Bergen, R. Hingorani, and S. Peleg. A 3-frame alorithm for estimating 2-component image motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:886–896, 1992.

[5] I. Biederman. Recognition by components. *Psychological Review*, 94:115–147, 1987.

[6] T. J. Broida and R. Chellappa. Estimation of object motion parameters from noisy images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:90–99, 1986.

[7] J. F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, November 1986.

[8] I. Daubechies. *Ten Lectures on Wavelet.* SIAM, Philadelphia, PA, 1992.

[9] S. Floyd, V. Jacobson, C. Liu, S. McCanne, and L. Zhang. A reliable multicast framework for light-weight sessions and application level framing. In *ACM SIGCOMM*, pages 342–356, 1995.

[10] B. Furht, S. Smoliar, and H. Zhang. *Video and Image Processing in Multimedia Systems.* Kluwer Academic Publishers, Norwell, Massachusetts, 1995.

[11] D. Huttenlocher and P. Wayner. Finding convex groupings in an image. *International Journal of Computer Vision*, 8:7–29, 1992.

[12] S. McCanne and V. Jacobson. vic: A flexible framework for packet video. In *ACM Multimedia*, pages 511–522, 1995.

[13] B. Parvin and G. Medioni. A dynamic system for object description and correspondence. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 393–399, 1991.

[14] B. Parvin, C. Peng, W. Johnston, and M. Maestre. Tracking of tubular molecules for scientific applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:800–805, 1995.

[15] B. Parvin, S. Viswanatha, and U. Dahmen. Tracking of convex objects. In *Int. Symp. on Computer Vision*, 1995.

[16] Bellman R. *Dynamic Programming.* Princeton University Press, 1957.

[17] A. Sha'ashua and S. Ullman. Structural saliency: the detection of globally salient structures using a locally connected network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 321–327, Tampa, FL, 1988.

[18] G.S. Young and R. Chellappa. 3-d motion estimation using a squence of noisy stereo images: Models, estimation, and uniqueness results. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:735–759, 1990.

[6] $0 \leq \gamma \leq \pi$